Studying Software Developer Expertise and Contributions in Stack Overflow and GitHub

Technical Report

Sri Lakshmi Vadlamani and Olga Baysal

Department of Computer Science Carleton University, Ottawa

srivadlamani@cmail.carleton.ca
olgabaysal@cmail.carleton.ca

07 April 2020

1. INFORMAL OVERVIEW

The paper seeks to understand whether experts in software development are made based on their knowledge and experience alone or there are other ingredients that make them "experts". It was found out that both knowledge and experience are only necessary but not sufficient and that soft-skills are the key in making them an "expert" in the field of software development.

An exploratory survey of software developers was conducted and a combination of quantitative and qualitative methods were used to analyze the survey results. Based on our criteria of "active users" a total of 423 software developers were identified as being active on both Git Hub and Stack Overflow. A survey questionnaire was sent to these users and of this 73 started working on the survey and 41 finally completed the survey. The study aimed to understand, classify and analyze the underlying themes and concepts that could be derived from these survey responses.

The completed responses were split into 324 quotations or individual statements and an open card sort was performed. The objective of this was gain insight into high-level themes and/or categories about expertise. Five (5) open ended questions were posed and during the qualitative analysis phase, 4 high-level 'concept-categories' emerged from these:

1. Expert's skills (participants – 41; quotes – 176; categories – 23)
2. Contribution in GitHub (participants – 41; quotes -75; categories – 9)
3. Contribution in Stack Overflow (participants – 41; quotes – 50; categories – 9)
4. Tags in SO match the expertise (participants – 41; quotes – 23; categories – 5)

An open coding technique was applied on the 324 comments to classify them into natural evolving 'concept-categories'. Moreover, the high-level 'concept-categories' were further sub-divided into 46 categories or themes representing various aspects of expertise and each category consisted of between one (1) and forty-four (44) comments or quotations. An overview of the concept categories and their themes are elaborated in the next section with details of the participant and the number of quotations (statements) in each 'concept-category' and theme. Moreover, for each theme a synthetic quote (SQ), an overall summary of the thought on that specific topic, is derived from the overall quotations or statements made on the specific theme. For a complete analysis, please refer to Appendix A.

2. CONCEPT CATEGORIES

The four (4) 'concept-categories' that have emerged during the card sorting exercise do not necessarily have direct correspondence to the tasks performed by experts on the social coding platforms. Instead, these categories are a combination of actions and mental activities performed by experts contributing to the social coding platforms (GH and SO). The four (4) concept categories are:

1. **Defining features of an expert** (*Software developer skills; Monitoring skills*). This area pertained to identifying the essential skills and competencies of experts who contributed to software development, in general (specifically within social coding platforms) and the features that an expert would have to possess and how this could be monitored.
2. **Factors driving experts to contribute in GitHub** (*contribution in GitHub)*. This concept category offers insights into the reason's experts contribute to GitHub and the associated inhibiting factors.
3. **Factors driving experts to contribute to Stack Overflow** *(contributions in Stack Overflow*). This concept category offers insights into the reason's experts contribute to Stack Overflow and the associated inhibiting factors.
4. **Challenges faced by experts in contributing to online collaborative platforms** (*tags in Stack Overflow match the expertise*). This is very specific to Stack Overflow, as it is the only platform amongst the two that allow for tags. This 'concept-category' checked on how the tags in Stack Overflow match the expertise of a software developer.

Technical skills are critical for a programmer to possess; for example, under the software developers' skills question, based on the card sorting exercise, there were 44 quotes that were classified as 'technical' and this was from comments obtained from 21 out of the total respondents who believed that this is a key skill. It must be noted that some participants emphasized by providing more than one quote or comment related to this specific theme. However, soft skills like communication skills (P – 15; Q – 15), analytical thinking (P – 24; Q – 26), planning & organization (P – 9; Q – 11) and behavior that includes tenacity, patience and perseverance (P – 9; Q – 17) play a crucial in forming expertise. Contribution to GH and SO does have very similar themes, in terms of the contribution of experts. However, from the qualitative analysis, some subtle difference can be observed. For example, "*self-needs*" is the most crucial driving factor for expert's contribution to GH (P – 27; Q – 34); whereas, this factor is not very important for contribution to SO (P – 3; Q - 3). On the other hand, "skills/expertise" are crucial for contribution to SO (P – 19; Q -20). Finally, tags within SO are automatically selected and do not often reflect the current state or expertise of a contributor (P – 7; Q – 7).

Open Ended Questions in the Survey	# of Participants	# of Quotes
1. Software Developers' Skills	**41**	**153**
▪ Communication skills	15	15
▪ Technical	21	44
▪ Analytical thinking	24	26
▪ Planning & organizing	9	11
▪ Creativity	3	3
▪ Experience	2	4
▪ Understanding	4	4
▪ Critical thinking	3	3
▪ Behavior	9	17
▪ Vision	1	1
▪ Domain knowledge	2	2
▪ Broad knowledge	7	8
▪ Requirements	6	7
▪ Teamwork	3	3
▪ Teaching skills	2	2
▪ Searching/seeking help	2	2
▪ Miscellaneous	1	1

2. Monitoring Skills	**14**	**24**
▪ Self-tracking issues	3	3
▪ Dashboard	5	7
▪ Online collaborative platforms	3	4
▪ Online documentation	1	1
▪ Testing tools	5	6
▪ Time tracking tools	3	3
3. Contribution to GitHub	**41**	**75**
▪ Hobby/fun	7	8
▪ Issue complexity	6	6
▪ Organization culture	1	2
▪ Self needs	27	34
▪ Job	1	1
▪ Helping	5	5
▪ Expertise	10	10
▪ Open source project	1	1
▪ Lack of time	8	8
4. Contribution in Stack Overflow	**41**	**50**
▪ Better answer	3	3
▪ Helping	3	3
▪ Hobby/fun	5	5
▪ Skill/expertise	19	20
▪ Self-learning	3	3
▪ Own needs	6	6
▪ Alternative solutions	1	1
▪ Tagging system	7	8
▪ Rewarding system	1	1
5. Tags in SO match the expertise	**20**	**23**
▪ Limited participation	1	1
▪ Limitation of platform	1	1
▪ Gamification configuration	7	7
▪ Mismatch if tags with expertise	4	6
▪ Miscellaneous	8	8

An appendix to this document, provides a comprehensive overview of the categories and themes, and a selection of participants' quotes. Furthermore, a synthetic quote (SQ) synthesizing participants' comments/ statements for each theme is provided in the appendix.

3. ABOUT THE AUTHORS

Sri Lakshmi Vadlamani is a master student in computer science with data science specialization at the School of Computer Science, Carleton University. Her research interests include mining software repositories, topic classification and topic modeling using natural language processing. She has a MS in Electrical Engineering from California State University, Northridge.

Olga Baysal is an Associate Professor at the School of Computer Science, Carleton University. Her research interests span a wide range of software engineering, including empirical software engineering, mining software repositories, data science, data analytics, software analytics, software maintenance and evolution, and human aspects of software engineering. She obtained her PhD from the University of Waterloo, Canada.

Appendix A

Software Developers Skills
[Participants (P): 41; Quotes (Q): 153]

- ### [P: 15, Q: 15] Communication skills
  - **Includes**: Effective & clear communication is a key skill for a software developer.
  - **Summary Quotient (SQ)**: Good communication skills defines software developer's expertise and are important for the below reasons:
    - Discuss technical options with team members and colleagues, ensuring that point is conveyed appropriately.
    - Teach a colleague or friend about the great technology/practice which are being used, i.e., effective knowledge transfer.
    - Defend the development or technical strategy that one believes is best while conveying to manager/customer that it has more pros than cons... And describing accurately those pros and their advantages as well as the consequences of not doing it right.

- ### [P: 21, Q: 44] Technical
  - **Includes**: Software developer's technical skills like programing languages, version control, testing, debugging, designing, documentation.
  - **SQ**: It is important to have strong understanding on the programing languages, data structures, debugging or algorithms and to be able to interpret ideas clearly into code, to be an expert in software development.
    - P1Q21P1. Proficiency in the particular programming language.
    - P6Q21P5. writing clear idiomatic programs in at least one language.
    - P25Q21P1. "Thinking in code" (fluently translating ideas into code).

- ### [P: 24, Q: 26] Analytical thinking
  - **Includes**: Problem solving skills and ability to think abstractly.
  - **SQ**: Being able to think logically and breaking down of complex problems to simpler problems would add to the definition of software developer's expertise.
    - P16Q21P2. reducing problems to minimal test cases
    - P32Q21P2. big-picture/small-details analytics
    - P6Q21P4. Breaking complex problems into simpler tasks

- ### [P: 9, Q: 11] Planning & organizing
  - **Includes**: Organizing, focusing and planning.
  - **SQ**: Being able to balance the workload and achieve overall goals of time and space performance are defined to be important skills
    - P16Q21P4. balancing time/space trade-offs
    - P41Q21P2. being organised

- **[P: 3, Q: 3] Creativity**
  - **Includes**: Improvising and having aesthetic sense.
  - **SQ**: Creativity is the source to improvise solutions to problems for dominating complex systems such as software development.
    - P2Q21P2. Improvisation
    - P5Q21P2. Aesthetic sense

- **[P: 2, Q: 4] Experience**
  - **Includes**: Breadth of knowledge, experience.
  - **SQ**: Having the breadth and depth of knowledge and experience to back it up is very important. In other words, clear representation of the code comes with experience, which is key for an expert software developer.
    - P16Q21P1. writing code in a way that can be understood by others
    - P16Q21P6. (or preferably working with methods that are "correct-by-construction")

- **[P: 4, Q: 4] Understanding**
  - **Includes**: Common sense.
  - **SQ**: Having the capacity to comprehend dynamic ideas and express them in simple/clear terms is an important skill.
    - P14Q21P3. ability to understand abstract concepts
    - P39Q21P1. expressing intent clearly

- **[P: 3, Q: 3] Critical thinking**
  - **Includes**: Flexibility and pragmatism.
  - **SQ**: It is very important to actively & skillfully conceptualize, apply and analyze the problem and solve it.
    - P24Q21P4. Quickly determining what is relevant
    - P8Q21P1. pragmatism

- **[P: 9, Q: 17] Behavior**
  - **Includes**: Tenacity, patience, continuous learning, curiosity & flexibility.
  - **SQ**: Behavior and attitude towards problem solving is a very crucial skill and this requires a programmer to be tenacious, patient, curious and flexible in order for them to reach their goals. This also requires them to show ample of empathy towards others both at their work place and outside.
    - P41Q21P1. Being empathetic about the needs of other people
    - P20Q21P5. Workplace behavior
    - P13Q21P4. Tenacity

- **[P: 1, Q: 1] Vision**
  - **Includes**: Attention to the end goal.
  - **SQ**: Having a clear and a well-defined end goal is very important in completing a project.
    - P8Q21P3. Pay attention to the goal

- **[P: 2, Q: 2] Domain knowledge**
    - **Includes**: Domain knowledge.
    - **SQ**: Key to solving a problem is to both understand the problem and to have a good understanding of the domain where the software is being used and/or the domain of the programming.
        - P1Q21P3. understanding of the domain that is being worked in (not programming itself, but the domain that the software is being used in)
        - P20Q21P3. Domainknowledge

- **[P: 7, Q: 8] Broad knowledge**
    - **Includes**: Comprehensiveness, knowledge ecosystem, self-learning, fundamentals.
    - **SQ**: Having a wider knowledge base that is outside of programming itself would help; it would help the programmer in gaining solid fundamental understanding of the issues at hand. This however, requires substantial self-learning, researching on the subject and having an ability to tap into the knowledge ecosystem. This also includes knowledge and good grasp of other languages: Mathematics and English.
        - P7Q21P1. Comprehensiveness
        - P16Q21P3. researching prior art
        - P20Q21P4. Ecosystemknowledge
        - P21Q21P3. self-learning
        - P33Q21P2. fundamentals

- **[P: 6, Q: 7] Requirements**
    - **Includes**: Requirements gathering, defining specification and translate requirements to design solution.
    - **SQ**: The first step of any project is requirement gathering of the end-user; clearly defining the specifications of the inputs and most importantly, it requires an ability to translate the requirements into solutions.
        - P10Q21P1. requirements elicitation
        - P11Q21P2. Translate requirements to design solutions
        - P25Q21P3. attention to detail
        - P33Q21P1. knowing what needs to be done
- **[P: 3, Q: 3] Teamwork**
    - **Includes**: Teamwork and ability to work together.
    - **SQ**: Key to any project (including software development) success is the ability to work in a team and to understand and appreciate the contribution of others. This comes from a basic recognition that all projects (of some value and size) need contribution of other stakeholders.
        - P22Q21P3. ability to work together
        - P20Q21P2. teamwork

- **[P: 2, Q: 2] Teaching skills**
    - **Includes**: Teaching and mentoring.

    - **SQ**: One of the key skills in software development is to teach and mentor others. This would create an effective eco-system of helpful users and builds a support network, which in turn would make the project more efficient and effective.
        - P24Q21P2. Mentoring others;
        - P18Q21P4. Teaching users

- **[P: 2, Q: 2] Searching/seeking help**
    - **Includes**: Google error messages.
    - **SQ**: Self-help sometimes is the best help as it might be difficult to communicate pain points of a developer to others, in that case, it is recommended to rely upon internet for possible solutions and guidance to solve the problem.
        - P24Q21P3. Googling error messages
        - P28Q21P4. Searching solution on Internet
- **[P: 1, Q: 1] Miscellaneous**
    - **Includes**: Cutting and pasting.
    - **SQ**: There is one irrelevant comment and hence is classified as “garbage”.
        - P26Q21P4. cutting-and-pasting

Monitoring Skills [Participants (P): 14, Quotes (Q): 23]

- **[P: 3, Q: 3] Self tracking Issues**
    - **Includes**: Monitoring resolved issues, having a database of public facing issues.
    - **SQ**: After execution of a project (eg., software development), the monitoring and control phase is an important one within the overall project. Hence, creating an issue-register and a comprehensive database of all the issues and their lessons learned is a key to overall project success.
        - P32Q23P1. I mostly reflect on what I am doing, why, and if it helps me solve problems faster and maintain the code in the long run.
        - P16Q23P3. public-facing issue databases
        - P35Q23P1. Monitoring issues resolved

- **[P: 5, Q: 7] Dashboard**
    - **Includes**: GitHub dashboard with commit graph, tracking metrics, heat charts, GitHub insights.
    - **SQ**: Providing summary information is essential to understand and easy to communicate with other stakeholders. Social platforms, like GitHub, provide dashboards to summarize the contribution and progress of an individual developer. These dashboards provide valuable information both to the developer as well as other relevant stakeholders
        - P5Q23P1. via github insight
        - P20Q23P1. GitHub dashboard with commit graph

        - P38Q23P1. tracking metrics I'm interested in
        - P30Q23P2. GitHub/GitLab heat charts

- **[P: 3, Q: 4] Online collaborative platforms**
    - **Includes**: GitHub, OhLoh, Git Commits.
    - **SQ**: Monitoring tools are provided within the online collaborative platforms to help track progress.
        - P7Q23P1. git commits
        - P30Q23P4. OhLoh, etc.

- **[P: 1, Q: 1] Online documentation**
    - **Includes**: Publicly available progress reports.
    - **SQ**: As part of project monitoring and audit it is critical to have publicly available progress reports.
        - P16Q23P2. public-facing progress reports

- **[P :5, Q :6] Testing tools**
    - **Includes**: Forensics, continuous integration and testing, linters, static analysis.
    - **SQ**: Testing is a key part of the monitoring and control process of a project. Testing could be a continuous quality check and/or it could be done at the end of the project before handing over the project (software program) to the end user.
        - P39Q23P1. Automated code metrics, such as cyclometric complexity or code coverage percentages.
        - P16Q23P1. continuous integration and testing
        - P14Q23P2. static analysis (like Sonar cloud)
        - P10Q23P3. Forensics
        - P10Q23P1. Linters
        - P29Q23P1. Tests

- **[P: 3, Q: 3] Time tracking tools**
    - **Includes**: Time tracking with detailed activity breakdown, burn-down charts.
    - **SQ**: Time is always of the essence. It is essential to keep a good/detailed track of time on each activity and sub-activity. This would help know/predict well in advance any time overruns. Hence, time tracking tools is key within the monitoring and control process of the project. Also included in here is an altruistic theme of helping others and thus a more well-rounded growth of the development team.
        - P3Q23P1. Time tracking with detailed activity type breakdown
        - P10Q23P4. burn-down charts
        - P35Q23P2. time spent helping others

Contribution in GITHUB
[Participants (P): 41, Quotes (Q): 75]

- **[P: 7, Q: 8] Hobby/fun**
  - **Includes**: Hobby, personal interest, fun, enjoy coding.
  - **SQ**: Contribution on GitHub is not always driven by professional requirements and choices. In fact, there is ample evidence that notes that contribution to social coding platforms is often driven by software developers' personal interest and/or hobby. They contribute because they enjoy working on these challenges created by projects.
    - P4Q25P2. some of it is hobby though
    - P11Q25P1. Personal interest. If I spend my time for free, I want to enjoy it
    - P25Q25P3. I prefer to contribute to projects that use languages I enjoy working in
    - P37Q25P1. Has to be interesting

- **[P: 6, Q: 6] Issue complexity**
  - **Includes**: Complexity of the project, size of the issue and mentally challenging.
  - **SQ**: Contribution to GitHub is often related to the complexity of the project, size of the issues and intrigue and challenge involved. Contribution is generally positively correlated with the level of challenge.
    - P23Q25P2. When I try to contribute, I generally choose small intriguing issues.
    - P37Q25P2. mentally challenging.
    - P8Q25P1. It mostly depends on the complexity of the project.
    - P7Q25P2. size of the issues.
    - P30Q25P1. As a Linux distribution contributor most of my contributions are based on issues packaging software.

- **[P: 1, Q: 2] Organization culture**
  - **Includes**: Project organization and process of building.
  - **SQ**: Contribution to GitHub requires a culture of project management that involves detailed project organization that involves planning, execution and monitoring during the process of building a software development.
    - P7Q25P3. build process
    - P7Q25P4. organization of the project

- **[P: 27, Q: 34] Self needs**
  - **Includes**: Driven by own needs and scratch my own itches.
  - **SQ**: Contribution to social coding platforms often begins with a problem or challenge the one is faced with and do not have ready available answers. In such instances, developers often rely on the social platforms for ready answers; however, if they do not find one

they tend to work on developing the same and thus end up contributing to the overall knowledge base.

        - P1Q25P3. a feature I need I will try and contribute
        - P1Q25P1. Based off my own needs.
        - P3Q25P2. contribute bug fixes to software I use
        - P10Q25P2. I need to fix something broken that's blocking me
        - P13Q25P1. I generally only contribute bug fixes if I run into something in a third-party library.
        - P17Q25P1. If I encounter a bug and I want it fixed badly enough, I will attempt to and if successful contribute my fix
        - P21Q25P1. What I'm currently using that needs changes
        - P27Q25P1. I use the project and want it to keep being maintained
        - P34Q25P1. I contribute to things that I use and
        - P41Q25P1. Ones that I use and that have something that's causing me problems which I'm able to fix fairly easily.
        - P35Q25P1. I generally contribute to a project that I will be using in some way or another

- **[P: 1, Q: 1] Job**
    - **Includes**: Paid work.
    - **SQ**: Only a small percent of respondents noted that they contribute to social coding platforms as part of their work.
        - P3Q25P3. paid work.

- **[P: 5, Q: 5] Helping**
    - **Includes**: Useful to others, good quality help.
    - **SQ**: Contribution to social coding platforms often is due to altruistic reasons. Some developers want to help others with a part of their code or challenge that they are stuck on; or the contributors help others to try and make others' code more efficient.
        - P6Q25P2. I contribute to new projects I think would be useful to others
        - P8Q25P2. how much time I have to invest so I can contribute good quality help.
        - P10Q25P1. I contribute because I care about what's being made,
        - P18Q25P1. They look like they might be useful.

- **[P :10, Q :10] Expertise**
    - **Includes**: Contribute to libraries to fix bugs, provides patches.
    - **SQ**: Contributions GitHub often involve experts who are currently working on similar fields in their current job (or in a professional setting) do contribute to certain libraries to fix bugs or provide important patches.
        - P4Q25P1. What I end up using for work guides what I contribute to mostly

        - P16Q25P2. but I will patch anything if I am able.
        - P29Q25P2. anything where I see a problem that I can find a decent solution to in an hour or so
        - P36Q25P2. Mostly contribute to libraries used in work-related projects
        - P32Q25P1. I contribute to projects on GitHub if I can fix bugs

- **[P: 1, Q: 1] Open source project**
    - **Includes**: B2B solutions or development-oriented solutions.
    - **SQ**: There is one comment on the availability of the project and its relation to contribution. Contribution to solutions depend on the open-ended nature of source of the project; as the contributors believe in making a base contribution to the solution that would allow others to build upon.
        - P2Q25P2. I work on B2B solutions or dev oriented solutions, so the core of these solutions can be open-sourced so other devs can contribute to it.

- **[P: 8, Q: 8] Lack of time**
    - **Includes**: Do not contribute due to time constraints, SO answerer community can be cruel to answer seekers.
    - **SQ**: There are a few irrelevant comments on why people do not contribute to GitHub. Some of them also declined to state the reasons and others noted that they do not contribute because of lack of time and even others noted that they feel intimidated by the answerers and hence are hesitant to ask questions in the community.
        - P13Q25P2. I have my own projects to work on on the side, so I don't really have the time or interest to contribute to other open source projects.
        - P23Q25P1. I rarely contribute because I don't have enough spare time.
        - P25Q25P4. The SO answerer community can be quite cruel to askers; this has made me less inclined to participate recently.

Contribution in Stack Overflow [Participants (P): 41, Quote (Q): 50]

- **[P: 3, Q: 3] Better answer**
    - **Includes**: Quest for better (optimal or efficient) solution to a problem.
    - **SQ**: Contributions in StackOverflow are sometimes because the developer was unable to find a suitable or efficient answer to their problem (question). Hence, they usually tend to go back, once they find a solution, and contribute a solution that they believe is better fit.
        - P17Q26P1 If I find someone who has asked the same question but the answers are not good, I like to return later to add my own answer once I have found it

        - P6Q26P2. answer questions I've had for which there wasn't a complete answer

- **[P: 3, Q: 3] Helping**
    - **Includes**: Helpful, useful and save time.
    - **SQ**: Contributors to SO usually contribute when they believe that their contribution would be help other users and it would save someone else's time.
        - P7Q26P1. Whenever I think my contribution would be useful to other users
        - P38Q26P3.it still will save someone's time

- **[P: 5, Q: 5] Hobby/fun**
    - **Includes**: Enjoy it as a past-time activity; interests the person and reflects a positive mental state.
    - **SQ**: Contributors often contribute on SO because they enjoy doing so and because it interests them and it reflects their mind-set of helping others. Usually, they also do it as a past-time activity.
        - P31Q26P1. depends on my mental state
        - P9Q26P1. It interests me
        - P3Q26P1. All over the board, as a pastime

- **[P: 19, Q: 20] Skill/expertise**
    - **Includes**: Familiar programming languages and knowledge area where one feels qualified to answer.
    - **SQ**: Contributors on SO often contribute on programming languages and libraries that they know are qualified and have a certain level of expertise to answer the questions. Furthermore, contributors often contribute in areas that they believe are niche and there is not much work or contribution already done in such area.
        - P1Q26P1. Based on programming languages or libraries that I know and
        - P4Q26P1. I usually try to answer some questions in an area I know something about
        - P12Q26P1. the one I feel qualified
        - P16Q26P1. Again due to time constraints: I recently use Stack Overflow primarily as a way of documenting my own hard-won findings on a topic (e.g. posting and answering a question, then linking to it in the code), but am happy to get other answers and opinions added.
        - P20Q26P1 Needs to be a niche area where there isn't much expert competition. I can't compete with ppl writing a comprehensive answer in &lt;10s ;)
        - P25Q26P1. I participate in topics I know well, especially where I have niche expertise.
        - P37Q26P1. I participate on discussions I can contribute. I avoid discussions in which I can't.

- **[P: 3, Q: 3] Self learning**
    - **Includes**: Testing abilities, learning about technologies that one is interested and effectively sharing knowledge.
    - **SQ**: Contributors on SO are motivated by self learning and desire to share their knowledge with others and this drives their level and extent of contribution. Also, they usually tend to participate in technologies that they would like to learn and currently they do not have the bandwidth or chance to work on in their regular jobs.
        - P16Q26P2. When I had more time I participated in C++ to learn about the C++11 standard and test my ability to share that knowledge effectively.)
        - P32Q26P2.I might also ask questions and immediately answer a question if I've just found a solution to a tricky question.
        - P8Q26P1. I mostly participate in questions related to technologies that I want to learn but I don't have currently chance to work on at work so I can try to help solving real life problems with that technology.

- **[P: 6, Q: 6] Own needs**
    - **Includes**: Work based on personal needs and relevance to personal work.
    - **SQ**: Contributors often monitor SO to get answers for questions pertaining to their own projects. In essence, their activity on SO depends on the relevance to their current work or if their search on internet is not providing them necessary answers.
        - P35Q26P1. I monitor stack overflow for questions related to my open source project. I seldom answer other questions.
        - P14Q26P1. Based on relevance to current work
        - P19Q26P1 If I land on the question during a search
        - P22Q26P1 Missing key information on the internet
        - P36Q26P1. Only participate in topics that relate to my own work

- **[P: 1, Q: 1] Alternative solutions**
    - **Includes**: New perspective.
    - **SQ**: There is one comment on this aspect, wherein the respondent has noted that their contribution to SO is to add “new perspective” or a new approach to the problem that they are attempting to solve.
        - P11Q26P2. that my opinion adds a new perspective

- **[P: 7, Q: 8] Tagging system**
    - **Includes**: Some irrelevant observations, viz., management changes, lack of time.
    - **SQ**: Some responses were not appropriate and hence been classified as ‘miscellaneous’. Some of the responses are below:
        - P23Q26P1. I don't partecipate, mainly because I don't have enough time.
        - P30Q26P1. I no longer contribute to SO due to their management changes this year.

        - P13Q26P1. I used to answer more questions when I had more free time at work. Now I'm busy enough that I don't search for more questions to help out with.
        - P10Q26P1. I mostly don't participate in Stack Overflow any more. It's hard to find interesting questions to answer,
        - P32Q26P1. I ask questions on Stack Overflow if I think a problem is well suited and clearly defined.

- [P: 1, Q: 1] Rewarding system
    - **Includes**: contribute when technically challenging; however, SO rewards contribution to easy topics.
    - **SQ**: There is one response with regards to complexity of contribution and the reward mechanism on SO. The respondent noted that they contribute when the problem is challenging and mentally stimulating. However, the person has noted that SO rewards contribution to easy topics, which need to be further investigated.
        - P34Q26P1. Questions that I can answer, particularly if technically challenging. Unfortunately SO rewards contributing to easy topics over hard.

Tags in Stack Overflow match the expertise [Participants (P): 20, Quotes (Q): 23]

- [P: 1, Q: 1] Limited participation
    - **Includes**: Respond only to popular questions.
    - **SQ**: Some participants only respond to popular queries.
        - P6Q29P1. I answered one very popular question with a very popular answer

- [P: 1, Q: 1] Limitation of platform
    - **Includes**: Only limited to specialized niche areas.
    - **SQ**: Some participants are hesitant to respond as they believe that SO only deals with specialized niche areas; anything outside of that is not well dealt with and hence they believe is a waste of time.
        - P33Q29P1. I do a lot more than I write on SO. SO is a waste of time outside specialized niches.

- [P: 7, Q: 7] Gamification configuration
    - **Includes**: Automatically selecting tags
    - **SQ**: Tags within SO are automatically selected and do not often reflect the current state or expertise of a contributor. Some examples:
        - P34Q29P2. My SO profile reflects expertise during PhD.

    - P26Q29P1. I have two very highly-voted answers that seem to outweigh all the other stuff I've done on SO: one on VLAs in C++ and one on strings/Unicode in C#.
    - P3Q29P1. Maybe historic skew, definitely skewed towards mainstream, popular tags
    - P17Q29P1. I am not so interested in answering tagged questions as it tends to be a race and often the site regulars will beat you to it, so you put in a lot of effort answering for little gain. By focusing on questions that do not have good answers yet, I feel I get more value out of contributing.
    - P18Q29P1. They reflect the tags that have the most questions asked.
    - P20Q29P1. As I said, my key expertise is in areas where there are so many experts that all questions are answered very quickly by others. My top rated answers are on niche areas where I'm one of the only experts.
    - P21Q29P1. Top tags metric doesn't work very well if you answered 2-3 questions many years ago that happened to have those tags on

- **[P: 4, Q: 6] Mismatch if tags with expertise**
  - **Includes**: Mismatch between tags assigned and the actual level of expertise.
  - **SQ**: Based on feedback, it is understood that sometimes there is a mismatch between tags assigned and the actual level of expertise of contributors.
    - P1Q29P1. I don't work professionally with Elm nor Haskell, and I'm at the beginner level for both.
    - P1Q29P2. Numpy and Django on the other hand, I do work professionally with and I am at the expert level for those.
    - P9Q29P2. my current expertise is mainly Go
    - P34Q29P1. In my day job I work with backend web tech (flask) and databases.
    - P35Q29P1. These are mostly related to the open source project I manage, not the daily work I have been doing for the past several years.

- **[P: 8, Q: 8] Miscellaneous**
  - **Includes**: Some miscellaneous observations: I don’t use SO; Don’t know my tags
  - **SQ**: There are some irrelevant comments and hence are classified as “miscallaneous”.
    - P2Q29P1. I don't use SO
    - P19Q29P1. I don't know what my tags are.
  - P27Q29P1. I found answers to questions that became popular because they concern a lot of people. Usually it's because I faced the problem, found a corresponding question on StackOverflow but the answers didn't help. After find the solution on my own, I would go back and add my answer.